



70320

UNIX 系统 V 第 4 版
程序员指南: X11/NeWS
图形窗口系统 XVIEW

UNIX® SYSTEM V
RELEASE 4

*Programmer's Guide: X11/NeWS®
Graphical Windowing System
XVIEW™*



UNIX Software Operation

電子工業出版社

目 录

前 言	(1)
0.1 先导	(1)
0.2 如何使用本手册	(1)
0.3 假设	(3)
0.4 本手册所用字体约定	(3)
0.5 有关的文档	(3)

第一章 XView 程序员模型	(5)
1.1 XView 程序员模型	(5)
1.2 面向对象的编程	(5)
1.2.1 对象类层次结构	(5)
1.2.2 对象的柄	(6)
1.3 基于属性的函数	(9)
1.3.1 创建和操纵对象	(9)
1.3.2 性质与属性	(9)
1.3.3 改变对象的属性	(10)
1.3.3.1 属性-值对	(10)
1.3.3.2 属性和值	(10)
1.4 对象的类型	(11)
1.4.1 通用对象	(12)
1.4.2 窗口对象	(13)
1.4.3 框和子框	(13)
1.4.4 子窗口	(14)
1.4.4.1 画布子窗口	(14)
1.4.4.2 正文子窗口	(15)
1.4.4.3 控制面板	(15)
1.4.4.4 菜单	(15)
1.4.4.5 滚动条	(16)
1.4.4.6 图标	(16)
1.4.5 不可视对象	(16)

1.5 通告器模型	(17)
1.5.1 回调风格的编程	(17)
1.5.2 为什么需要基于通告器的系统?	(18)
1.5.3 通告器、对象和应用程序的关系	(18)
1.5.4 直接调用通告器	(20)

第二章 创建 XView 应用程序

(21)

2.1 创建 XView 应用程序	(21)
2.2 界面概述	(21)
2.2.1 编译 XView 应用程序	(21)
2.2.2 XView 库	(21)
2.2.3 前导文件	(22)
2.2.4 命名约定	(22)
2.2.4.1 保留名字	(23)
2.2.5 XView 风格编程的例子	(23)
2.3 初始化 XView	(25)
2.4 创建和修改对象	(26)
2.4.1 使用 <code>xv_create()</code>	(26)
2.4.2 使用 <code>xv_find()</code>	(28)
2.4.3 使用 <code>xv_destroy()</code>	(29)
2.4.4 使用 <code>xv_set()</code> 和 <code>xv_get()</code>	(29)
2.5 <code>xv_main_loop()</code> 和通告器	(32)
2.6 登记事件处理器	(33)

第三章 通用和公共对象软件包

(35)

3.1 通用和公共	(35)
3.2 通用过程	(35)
3.3 公共过程和宏	(37)
3.4 属性	(39)
3.4.1 属性的次序	(40)
3.4.2 属性表	(40)
3.5 通用属性	(40)
3.6 公共属性	(40)

第四章 窗口对象软件包	(45)
4.1 窗口对象软件包	(45)
4.2 窗口过程和宏	(45)
4.3 窗口管理器过程和宏	(46)
4.4 属性	(47)
第五章 框	(59)
5.1 框	(59)
5.2 框的类型	(59)
5.2.1 窗口管理器的作用	(60)
5.3 过程与宏	(60)
5.4 框属性	(61)
5.5 基框属性	(63)
5.6 命令框属性	(64)
第六章 画布和开窗对象	(65)
6.1 画布和开窗对象	(65)
6.1.1 画布、视图和绘画窗口	(65)
6.2 创建画布	(66)
6.3 画布过程和宏	(66)
6.4 开窗过程和宏	(67)
6.5 画布属性	(68)
6.6 开窗属性	(71)
第七章 控制面板	(75)
7.1 控制面板	(75)
7.2 创建控制面板	(76)
7.2.1 创建控制面板项	(76)
7.3 过程和宏	(77)
7.4 控制面板区属性	(79)
7.5 控制面板项属性	(82)

7.6 控制面板按钮项属性	(88)
7.7 控制面板选择项和乒乓项属性	(88)
7.8 控制面板表项属性	(90)
7.9 控制面板消息项属性	(94)
7.10 控制面板滑块项属性	(94)
7.10.1 显示滑块	(96)
7.11 控制面板正文项属性	(96)

第八章 正文子窗口

8.1 正文子窗口	(99)
8.2 创建正文子窗口	(99)
8.3 设置正文子窗口属性	(99)
8.4 过程和宏	(99)
8.5 属性	(105)

第九章 TTY 子窗口

9.1 TTY 子窗口	(113)
9.2 创建 TTY 子窗口	(113)
9.3 过程与宏	(113)
9.4 属性	(114)

第十章 滚动条

10.1 滚动条	(117)
10.2 创建滚动条	(117)
10.3 过程和宏	(117)
10.4 属性	(118)

第十一章 菜单

11.1 菜单	(121)
11.1.0.1 菜单类型	(121)
11.1.0.2 弹出菜单	(121)
11.1.0.3 下拉菜单	(121)

11.1.0.4 右拉菜单	(122)
11.1.1 菜单项	(122)
11.1.1.1 选择项	(122)
11.1.1.2 互斥项	(122)
11.1.1.3 非互斥项	(122)
11.1.2 创建菜单	(123)
11.1.2.1 创建菜单项	(123)
11.2 过程和宏	(124)
11.3 菜单属性	(125)
11.4 菜单项属性	(132)

第十二章 通知

12.1 通知	(137)
12.2 创建和显示通知	(137)
12.3 过程和宏	(139)
12.4 属性	(140)

第十三章 光标

13.1 光标	(143)
13.2 创建光标	(143)
13.3 过程和宏	(144)
13.4 属性	(144)

第十四章 图标

14.1 图标	(147)
14.1.1 创建图标	(147)
14.2 属性	(147)

第十五章 不可视对象

15.1 不可视对象	(149)
15.2 显示	(149)
15.3 屏幕对象	(149)

15.4	SERVER(服务器)软件包	(150)
15.4.1	创建服务器(建立连接)	(150)
15.4.1.1	默认服务器	(150)
15.4.2	取得服务器	(151)
15.4.3	服务器图象	(151)
15.4.3.1	创建服务器图象	(151)
15.5	FULLSCREEN(全屏幕)软件包	(152)
15.6	服务器属性	(153)
15.7	服务方图象属性	(154)
15.8	屏幕属性	(156)
15.9	全屏幕属性	(156)
第十六章	字体	(161)
16.1	字体	(161)
16.2	创建字体	(161)
16.3	属性	(162)
第十七章	选择服务	(165)
17.1	选择服务	(165)
17.1.1	XView 选择模型	(166)
17.1.2	使用选择服务	(166)
17.2	过程和宏	(167)
17.3	属性	(172)
17.4	高级选择属性	(174)
第十八章	通告器	(177)
18.1	通告器	(177)
18.2	通告器如何工作	(177)
18.2.1	交互作用的类型	(178)
18.3	事件处理	(179)
18.3.1	子进程控制事件	(179)
18.3.1.1	回收死进程	(180)
18.3.1.2	进程的结果	(180)

18.3.2	输入-挂起事件(管道)	(181)
18.3.3	信号事件	(183)
18.3.3.1	对 signal() 的代替	(183)
18.3.3.2	异步事件处理	(186)
18.3.4	超时事件	(186)
18.3.4.1	周期反馈	(186)
18.3.4.2	轮询	(188)
18.3.4.3	检查间隔定时器	(188)
18.3.5	委托方事件	(188)
18.3.5.1	邮寄	(189)
18.4	插入	(189)
18.4.1	插入的使用	(190)
18.4.2	插入的接口	(190)
18.4.3	监视框的状态	(191)
18.4.4	插入缩放尺寸事件	(193)
18.4.5	修正框的破坏	(193)
18.4.5.1	插入委托方删除处理器	(195)
18.5	通告器控制	(196)
18.5.1	向 XView 移植程序	(197)
18.5.2	显式调度	(197)
18.5.2.1	隐式调度	(197)
18.5.3	退出	(198)
18.6	出错处理	(199)
18.6.1	排错	(199)
18.6.2	限制	(200)
18.6.2.1	要避免的系统调用	(200)
18.6.2.2	要避免的信号	(201)
18.7	高级通告器用法	(202)
18.8	通告	(203)
18.8.1	委托方事件	(203)
18.8.1.1	输出已完成事件	(204)
18.8.1.2	异常出现事件	(204)
18.8.2	获取事件处理器	(205)
18.9	更多的关于插入	(206)
18.9.1	登记插入器	(207)
18.9.2	调用下一个函数	(208)

18.9.3 删除插入函数	(210)
18.10 邮寄	(211)
18.10.1 委托方事件	(211)
18.10.1.1 投递时间暗示	(212)
18.10.1.2 实际投递时间	(212)
18.10.1.3 带实参的邮寄	(213)
18.10.1.4 存储管理	(213)
18.10.2 邮寄删除事件	(214)
18.10.2.1 投递时间	(215)
18.11 优先权	(215)
18.11.1 提供优先器	(215)
18.11.1.1 调度事件	(217)
18.11.1.2 获取优先器	(218)
18.12 通告器控制	(218)
18.12.1 启动	(219)
18.12.2 停止	(219)
18.12.3 总体析构	(219)
18.12.4 调度表	(220)
18.12.4.1 调度委托方	(221)
18.12.4.2 获取调度器	(221)
18.12.5 委托方删除	(221)
18.13 出错码	(221)
18.14 异步调用通告器的限制	(223)
18.15 问题讨论	(224)

附录 A 数据类型	(227)
-----------------	-------

A.1 数据类型	(227)
----------------	-------

附录 B 事件处理	(235)
-----------------	-------

B.1 事件处理	(235)
----------------	-------

B.2 XView 事件	(235)
--------------------	-------

B.3 事件宏	(235)
---------------	-------

B.4 输入事件码	(236)
-----------------	-------

B.5 输入事件描述字	(239)
-------------------	-------

附录 C 命令行实参	(241)
------------------	-------

C.1 命令行实参	(241)
-----------------	-------

附录 D OPEN LOOK 用户界面一致性	(243)
------------------------------	-------

D.1 OPEN LOOK 用户界面一致性	(243)
-----------------------------	-------

D.2 XView 1.0 内不受支持的 1 级特性	(243)
----------------------------------	-------

D.2.1 键盘和鼠标的定制	(243)
----------------------	-------

D.2.2 输入焦点反馈	(244)
--------------------	-------

D.2.3 菜单默认设置	(244)
--------------------	-------

D.2.4 弹出框中的默认按钮	(244)
-----------------------	-------

D.2.5 求助	(244)
----------------	-------

D.2.6 窗口背景	(245)
------------------	-------

D.2.7 通知	(245)
----------------	-------

D.2.8 正文函数	(245)
------------------	-------

D.2.9 控制项	(245)
-----------------	-------

D.2.10 滚动清单	(246)
-------------------	-------

D.2.11 性质窗口	(246)
-------------------	-------

D.2.12 非活动控制功能	(247)
----------------------	-------

D.3 XView 1.0 内受支持的 2 级特性	(247)
---------------------------------	-------

D.4 XView 1.0 内不受支持的 2 级特性	(247)
----------------------------------	-------

图和表

图 1-1: 创建控制面板项的代码	(11)
-------------------------	------

图 1-2: 创建控制面板的代码	(12)
------------------------	------

图 1-3: 常规程序的控制流	(17)
-----------------------	------

图 1-4: 基于通告器程序的控制流	(18)
--------------------------	------

图 1-5: XView 应用程序的输入事件流	(19)
-------------------------------	------

图 2-1: quit.c 程序	(24)
------------------------	------

图 2-2: xv_create() 函数调用	(26)
-------------------------------	------

图 2-3: 在控制区域中安装控制面板项	(28)
----------------------------	------

图 2-4: xv_set 和 xv_get 的定义	(30)
----------------------------------	------

图 2-5: 用单个调用来改变三个属性	(30)
---------------------------	------

图 2-6: <code>callback_example.c</code> 程序	(34)
图 12-1: <code>simple_notice.c</code> 程序演示通知提示的调用	(138)
图 18-1: 通告概貌	(178)
图 18-2: 插入的控制流	(190)
表 1-1: XView 的对象、软件包和数据类型	(7)
表 1-2: 通用(Generic)函数	(9)
表 2-1: 保留前缀	(23)
表 2-2: 对象和它们的属主	(27)
表 17-1: 选择正文和图形	(165)

前 言

0-1 先导

XView(基于 X 窗口系统的工作站视觉/集成环境)是一个用户界面工具包。它支持在 X 窗口系统下运行的交互作用的、基于图形的应用程序。这个工具包由 Sun Microsystems 公司和 AT&T 开发,它来源于更早些的 SunView™ 窗口系统的工具包。在工作站市场中,已有 2000 个以上的 SunView 应用程序,因此有很多程序员已经熟悉 SunView 应用程序员界面(API)。由于 XView 的存在,使 SunView 应用程序进入广阔的 X 市场成为可能。

对于在 X 中开发新应用程序的程序员来说,XView 有很多优点。XView 提供了成熟和行之有效的 API,后者是建于 Sun View 程序员开发经验之上的。XView 还有面向对象界面的特色,因而易懂、易学。

正如任何 X 工具包一样,XView 提供了一组预先建立起来的用户界面对象,如画布、滚动条、菜单和控制面板。这些对象的外观和功能遵守 OPEN LOOK 图形用户界面(GUI)规范,由 Sun Microsystems 和 AT&T 联合开发,作为 System V Release 4 的图形用户界面标准。它提供给用户一个简单的、一致的和高效的界面,以完成在应用程序中的各种任务。

XView 基于 Xlib。Xlib 是程序员可以使用的 X 窗口系统的最低层。尽管开发 XView 应用程序并不需要 Xlib 的编程经验,但是多了解一些 Xlib 是很有益的,当其应用程序要描绘图形时更有必要。

0-2 如何使用本手册

本手册从 XView 编程概述开始,接着是说明 XView 程序的构造的指导材料,后面的各章描述 XView 的各个软件包。

第一章:**XView 程序员模型**提供了作为面向对象编程系统的 XView 的概述,程序员创建和修改用以实现 OPEN LOOK 界面的对象。本章也讨论了作为接收事件的对象的窗口;还介绍了回调函数,用它作为登记面向应用的事件处理器的方法。

第二章:**创建 XView 应用程序**从应用程序开发者的观点开始,说明一个 XView 应用程序的基本元素,并叙述了在初始化 XView 和创建 XView 对象(如框和子窗口)所涉及的内容。

第三章:**通用和公共对象软件包**以调用的格式描述了创建、处理和删除 XView 对象所需的过程的基本集,也列举了通用的和公共的属性和过程。

第四章：**窗口对象软件包**叙述了由所有窗口对象(如画布、正文子窗口、控制面板和菜单)所共用的属性、过程和宏。

第五章：**框**说明了如何创建窗口框。有两种基本类型的框：基框和命令框。每个应用程序至少有一个基框，以管理子窗口、控制面板和其它对象。它提供了用来创建和管理框的各例程。

第六章：**画布和开窗对象**将画布作为子窗口或窗格的最基本类型。它提供了画布的模型。该模型允许绘图表面大于画布子窗口中可见的部分。

第七章：**控制面板**说明了 OPEN LOOK 的各种控制功能，这些功能是以控制面板的项目来实现的。它说明了如何创建和使用其上的按钮、检查盒、选项、表格、消息、乒乓功能、正文项和滑块。一组控制面板属性控制所有控制面板项和针对具体属性的公共行为。

第八章：**正文子窗口**叙述了如何创建正文子窗口和如何使用它的正文编辑功能。

第九章：**TTY 子窗口**叙述了完成终端仿真功能的两种子窗口：TERM 和 TTY。两者之间的主要区别在于 TERM 子窗口是可滚动的。

第十章：**滚动条**涉及滚动条的创建和使用。滚动条是贴附于另一窗口(如画布窗口、正文子窗口或操作面板)的窗口。滚动条软件包只管理滚动条。应用程序必须调整滚动对其窗口的影响。

第十一章：**菜单**说明如何实现菜单。用于控制面板的许多控制项也可用作菜单的项。

第十二章：**通知**说明弹出式窗口如何用作通知/对话框。

第十三章：**光标**示出了各种 OPEN LOOK 光标和它的使用。

第十四章：**图标**描述了位映象图用作应用程序的图标。当应用程序被关闭或被要求图标化时，应用程序将作为一个图标表示在屏幕上。

第十五章：**不可视对象**描述了一些隐藏对象：通用对象、服务器、屏幕、字体和全屏对象等。

第十六章：**字体**描述了如何在 XView 下描绘图形。本章叙述了两个基本方法：一种是使用 SunView pixwin 图形库，这对简单的绘图是方便的。另一种是使用 Xlib 绘图例程。

第十七章：**选择服务**讨论 XView 应用程序如何同另外的应用程序通讯。它说明 XView 如何提供按委托方向通讯协议手册 (ICCCM) 中所陈述的选择和其他功能。

第十八章：**通告器**叙述了通告器和高级的事件处理。它描述了通告器和 X 以及主机操作系统的关系。

附录 A：**数据类型**列举出由 XView 规定的数据类型。

附录 B：**事件处理**讨论事件类型和事件码。

附录 C：**命令行实参**列举出命令行实参的值和默认值。

附录 D：**OPEN LOOK 用户界面一致性**讨论 XView 与 OPEN LOOK GUI 规范的一致性。

0.3 假设

虽然本书对 X 编程时要用到的一些 C 语言不常用的特性提供了例子,但读者应该熟悉 C 语言。此外,通晓光栅图形学的原理也将是会有帮助的。

0.4 本手册所用字体约定

斜体字用于:

- 用户命令的选用项。
- 新术语被定义的地方。
- Xlib函数的实参^❶,因为它们可以像示出的那样键入,但并非一定要那样做。

等宽字体用于:

- 应逐字键入的代码,例如屏幕上的源码和正文。
- include文件的内容,如结构类型、结构成员、符号(规定常数和位标志)和宏。
- Xlib函数。
- 示例程序的子例程之名。

黑体字用于:

- 章节的标题。

0.5 有关的文档

本手册所提到的 OPEN LOOK GUI 功能规范和风格指南现在可从 AT&T 顾客信息中心(CIC)获得,在上班时间请拨文档咨询免费热线电话:1(800)432-6600,以索取下列文件:

- AT&T OPEN LOOK GUI Functional Specification and Style Guide,本手册的编目为:308-488。
- 有关XWIN和OPEN LOOK的所有其它文档。

^❶ 英文版为斜等宽字体,中文版因字体所限用斜体字表示。

第一章 XView 程序员模型

1.1 XView 程序员模型

XView 提供一组预先定义的用户界面成分给程序员，以简化他们在 X 窗口系统下应用程序的开发。这些成分实现了由 Sun Microsystems 公司和 AT&T 所开发的 OPEN LOOK 用户界面准则的“外观与感觉。” (“look and feel”)

本章为程序员提供一个 XView 的模型，在开始写 XView 应用程序之前了解这个模型是至关重要的。

1.2 面向对象的编程

对程序员来说，XView 是一个面向对象的系统。XView 的对象可以看成是建造块，应用程序的用户界面可由它们来组装成。每一块可以认为是来自某个软件包 (package) 的一个对象 (object)。从可用的一些软件包中选取某些对象，便可为应用程序建造起用户界面。

XView 遵守如下两个面向对象编程的基本原则：

- 对象以类层次结构来表示。
- 对象是不透明的数据模型。

1.2.1 对象类层次结构

XView 以树形结构来定义对象的类，例如，菜单是更为一般的窗口类的子类，后者又是绘图窗口 (drawable) 的子类，而绘图窗口 (与用户接口对象类一样) 是通用对象类 (Generic Object class) 的子类。每个类都有它的标识特性，使得它有别于另外的类或软件包。在 XView 中，类经常称为软件包，意指一组相关的功能元素。但是，有一些 XView 软件包，它们不是对象类层次结构的成员，如通告器 (NOTIFIER) 软件包。

某些对象是可视的，而另一些是不可视的。可视的对象包括窗口、滚动条、框、控制面板和控制面板项。不可视对象是这样的对象，它们本身没有外形，但是它们有辅助可视对象显示的信息。不可视对象的例子包括服务器、屏幕和字体对象。例如，屏幕提供诸如它能显示的颜色种类、或对象可能继承的默认前景和背景颜色的信息；字体则能提供显示正文的对象可用的字体信息。

XView 使用静态子类划分和链接继承作为其面向对象模型的一个部分。系统是可扩展的，用户可以创建新的类，它们可以基于也可不基于已存在的类。所有可视的和不可视的对象都是此对象分类系统的一个部分。

某个类的所有对象都继承其父类(也叫超类)的性质。通用对象 `XV_OBJECT` 软件包包含所有对象共享的某些基本性质。例如,为了优化存储,同一对象可以在屏幕的许多地方出现;为了保持此情况的记录,对象保存着它的实例的引用计数。如果对象的一个实例正在被创建或被删除,那么该情况就存入对象的状态域中。同样,由于所有的对象都有其父类,对象的父亲被存入对象的通用部分的域中。因为所有的对象都需要知道关于自身的信息,这些信息就被定义为通用对象(层次结构的顶层)的一部分。当某个对象需要针对某种特定的外观或功能得到更多的内容时,其较低层的类就规定相应的性质来实现它。

每个类都具有性质,它们为该对象的所有实例所共享。例如,各控制面板是(PANEL)(控制面板)软件包的一部分,该软件包具有某些性质;这些性质描述了它的布局(水平的或垂直的)或在控制面板上各个项(按钮)之间的间隔等内容;所有的控制面板共享这些性质,即使这些性质的状态可能是对象的不同实例。

如前所述, `XView` 使用了子类的方法,使得每个软件包可能继承它的超类的性质。`PANEL` 软件包是 `WINDOW`(窗口)软件包分出的子类,后者具有用于所有窗口的性质,诸如窗口尺寸、在屏幕上的位置、边框的粗细、深度、可视性和彩色映像信息等。因此, `PANEL` 软件包也可以动用 `WINDOW` 软件包所具有的那些性质。

`WINDOW` 软件包是根对象 `XV_OBJECT` 分出的子类(所有对象均如此),因此,控制面板可以设置诸如尺寸和位置等通用信息。

1-2-2 对象的柄

当用户创建一个对象时, `XView` 函数返回一个该对象的柄(handle)。以后,当用户希望操纵该对象或询问关于它的状态时,可将它的柄传送给适当的函数。这种依靠对象柄的方法是一种信息隐蔽(information-hiding)的方法。由于用户不能“看透”表示对象的实际数据结构,在这个意义上来说,柄是不透明的(opaque)。

每个对象类型都有相应的柄类型。由于C语言没有 opaque(不透明)类型,因此,所有上述的不透明数据类型都用 `typedef` 在 `XView` 的 `Xv_opaque` 或 `Xv_object` 类型中定义。

除不透明数据类型外,还有几个 `typedef`,它们不是定义指针而是定义事件(Event)、矩形(Rect)和矩形表(Rectlist)等结构。通常,指向这些结构的指针被传给 `XView` 函数,所以它们被声明为 `Event*`、`Rect*` 等。“*”不包括在 `typedef` 中的理由是这些结构是公共可用的。

表 1-1 列出了 `XView` 的所有对象、定义它们的软件包,以及对象的数据类型。